B 10

```
PPPPPPPPPPPP     HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
PPPPPPPPPPPP     HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
PPPPPPPPPPPP     HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
PPP         PPP  HHH         HHH  000         000  NNN         NNN  EEE
PPP         PPP  HHH         HHH  000         000  NNN         NNN  EEE
PPP         PPP  HHH         HHH  000         000  NNN         NNN  EEE
PPP         PPP  HHH         HHH  000         000  NNNNNN      NNN  EEE
PPP         PPP  HHH         HHH  000         000  NNNNNN      NNN  EEE
PPP         PPP  HHH         HHH  000         000  NNNNNN      NNN  EEE
PPPPPPPPPPPP     HHHHHHHHHHHHHHH  000         000  NNN   NNN   NNN  EEEEEEEEEEEE
PPPPPPPPPPPP     HHHHHHHHHHHHHHH  000         000  NNN   NNN   NNN  EEEEEEEEEEEE
PPPPPPPPPPPP     HHHHHHHHHHHHHHH  000         000  NNN   NNN   NNN  EEEEEEEEEEEE
PPP              HHH         HHH  000         000  NNN      NNNNNN  EEE
PPP              HHH         HHH  000         000  NNN      NNNNNN  EEE
PPP              HHH         HHH  000         000  NNN      NNNNNN  EEE
PPP              HHH         HHH  000         000  NNN         NNN  EEE
PPP              HHH         HHH  000         000  NNN         NNN  EEE
PPP              HHH         HHH  000         000  NNN         NNN  EEE
PPP              HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
PPP              HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
PPP              HHH         HHH     000000000     NNN         NNN  EEEEEEEEEEEEEEE
```

```
_$2

Val
---
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
000
```

```
**FILE**ID**LINKSUBS                                   E  3

LL            IIIIII    NN      NN  KK      KK    SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
LL            IIIIII    NN      NN  KK      KK    SSSSSSSS  UU      UU  BBBBBBBB      SSSSSSSS
LL              II      NN      NN  KK      KK  SS          UU      UU  BB      BB  SS
LL              II      NN      NN  KK      KK  SS          UU      UU  BB      BB  SS
LL              II      NNNN    NN  KK    KK    SS          UU      UU  BB      BB  SS
LL              II      NNNN    NN  KK    KK    SS          UU      UU  BB      BB  SS
LL              II      NN  NN  NN  KKKKKK        SSSSSS    UU      UU  BBBBBBBB      SSSSSS
LL              II      NN  NN  NN  KKKKKK        SSSSSS    UU      UU  BBBBBBBB      SSSSSS
LL              II      NN    NNNN  KK    KK            SS  UU      UU  BB      BB          SS
LL              II      NN    NNNN  KK    KK            SS  UU      UU  BB      BB          SS
LL              II      NN      NN  KK      KK          SS  UU      UU  BB      BB          SS      ....
LL              II      NN      NN  KK      KK          SS  UU      UU  BB      BB          SS      ....
LLLLLLLLLL    IIIIII    NN      NN  KK      KK  SSSSSSSS    UUUUUUUUUU  BBBBBBBB    SSSSSSSS        ....
LLLLLLLLLL    IIIIII    NN      NN  KK      KK  SSSSSSSS    UUUUUUUUUU  BBBBBBBB    SSSSSSSS        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %title 'LINKSUBS - Phone Link Subroutines'
0         module linksubs (
1                         ident='V04-000') = begin
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! Facility:     VAX/VMS Telephone Facility, Phone Link Subroutines
1 !
1 ! Abstract:     This module contains the subroutines necessary to support
1 !               the establishment and use of phone links, both local
1 !               and remote.
1 !
1 !
1 ! Environment:
1 !
1 ! Author: Paul C. Anagnostopoulos, Creation Date: 17 November 1980
1 !
1 ! Modified By:
1 !
1 !       V03-006 BLS0251         Benn Schreiber          8-Dec-1983
1 !               Modify mailbox name so that all numeric usernames
1 !               do not look like cluster device names.  Also allow
1 !               SS$_IVDEVNAM from $ASSIGN.  Convert to use $TRNLNM.
1 !
1 !       V03-005 PCA1020         Paul C. Anagnostopoulos 27-May-1983
1 !               Add check for shared memory mailboxes and signal fatal
1 !               error if present.
1 !
1 !
1 !       V03-004 MHB0088         Mark Bramhall           8-Feb-1983
1 !               Corrected underscore handling in PHN$ESTAB_LINK.
1 !
1 !       V03-003 MHB0087         Mark Bramhall           17-Jan-1983
```

```
1 !                     Added "pass through" routing capability.
1 !                     PHN$ESTAB_LINK now always displays its status via
1 !                     PHN$INFORM; its callers just check the returned status.
1 !
1 !     V03-002 PCA1004          Paul C. Anagnostopoulos  8-Nov-1982
1 !                     $ASSIGN no longer returns SS$_IVDEVNAM when assigning to
1 !                     a mailbox that doesn't exist.  It now returns SS$_NOSUCHDEV.
1 !
1 !     V03-001 PCA0041          Paul Anagnostopoulos    26-Mar-1982
1 !                     Major changes to convert from process name to user name.
1 !--
```

```
1 %sbttl 'Module Declarations'
1 !
1 ! Libraries and Requires:
1 !
1
1 library 'sys$library:starlet.l32';
1 require 'phonereq';
1
1 !
1 ! Table of Contents:
1 !
1
1 forward routine
1         phn$mbx_enable: novalue,
1         phn$estab_link,
1         phn$mbx_name: novalue,
1         phn$break_link: novalue,
1         phn$break_call: novalue,
1         phn$send_smb: novalue,
1         phn$force_links: novalue,
1         phn$forced_link: novalue;
1
1 !
1 ! External References:
1 !
1
1 external routine
1         phn$term_characteristic,
1         phn$cmp_target,
1         phn$fresh_screen,
1         phn$inform,
1         phn$kill_pub,
1         phn$make_pub,
1         phn$make_tsb,
1         phn$queue_smb,
1         phn$read_slave,
1         uns$net_connect: addressing_mode(general);
1
1 !
1 ! Own Variables:
1 !
1 bind
1         lnm_process_desc = $descriptor('LNM$PROCESS'),
1         lnm_job_desc = $descriptor('LNM$JOB'),
1         slave_task_desc = $descriptor('PHN$SLAVE_TASK_SPECIFIER');
```

```
%sbttl 'PHN$MBX_ENABLE - Enable Mailbox ASTs'
!++
! Functional Description:
!       This routine is called to enable an AST for our receive mailbox.
!       This allows us to be notified when some other process puts a
!       steering message into our mailbox.  The AST itself just creates
!       a standard steering message block from the message.  The message
!       has the following three parts:
!
!               1.      The 1-byte message type code.
!
!               2.      The complete node/user name string of the sender,
!                       followed by an eofrom character.
!
!               3.      Any additional message text, if required.
!
! Formal Parameters:
!       none
!
! Implicit Inputs:
!       global data
!
! Implicit Outputs:
!       global data
!
! Returned Value:
!       none
!
! Side Effects:
!
!--

global routine phn$mbx_enable: novalue = begin

own
        mbx_iosb: block[8,byte],
        mbx_buf: vector[phn$k_mbxsize,byte];

local
        status: long,
        op: ref pub;                    ! Pointer to our PUB.


! This internal routine is the AST handler.  We have to build a standard
! steering message block from the mailbox data.

routine mbx_ast: novalue = begin

local
        status: long,
        mbx_buf_dsc: descriptor;

! First we have to check the IOSB from the mailbox read.

check (.mbx_iosb[0,0,16,0]);

```

```
;  173          0499  3 ! Now we build a descriptor for the message text portion of the steering
;  174          0500  3 ! message.  We can then queue an SMB with the message type and text.
;  175          0501  3
;  176          0502  3 mbx_buf_dsc[len] = .mbx_iosb[2,0,16,0] - 1;
;  177          0503  3 mbx_buf_dsc[ptr] = mbx_buf[1];
;  178          0504  3 phn$queue_smb(.mbx_buf[0],mbx_buf_dsc);
;  179          0505  3
;  180          0506  3 ! Finally, we have to enable another read from our mailbox.
;  181          0507  3
;  182          0508  3 phn$mbx_enable();
;  183          0509  3 return;
;  184          0510  3
;  185          0511  2 end;


                                                                        .TITLE  LINKSUBS LINKSUBS - Phone Link Subroutines
                                                                        .IDENT  \V04-000\

                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                53  53  45  43  4F  52  50  24  4D  4E  4C  00000 P.AAB:  .ASCII  \LNM$PROCESS\                              ;
                                                            0000B         .BLKB   1
                                                 0000000B   0000C P.AAA:  .LONG   11                                         ;
                                                 00000000'  00010         .ADDRESS P.AAB                                     ;
                            42  4F  4A  24  4D  4E  4C  00014 P.AAD:  .ASCII  \LNM$JOB\                                  ;
                                                            0001B         .BLKB   1
                                                 00000007   0001C P.AAC:  .LONG   7                                          ;
                                                 00000000'  00020         .ADDRESS P.AAD                                     ;
5F  4B  53  41  54  5F  45  56  41  4C  53  24  4E  48  50  00024 P.AAF:  .ASCII  \PHN$SLAVE_TASK_SPECIFIER\                 ;
                        52  45  49  46  49  43  45  50  53  00033                                                            ;
                                                 00000018   0003C P.AAE:  .LONG   24                                         ;
                                                 00000000'  00040         .ADDRESS P.AAF                                     ;

                                                                        .PSECT  $OWN$,NOEXE,2

                                                            00000 MBX_IOSB:
                                                                        .BLKB   8
                                                            00008 MBX_BUF:.BLKB   256

                                                                  LNM_PROCESS_DESC=     P.AAA
                                                                  LNM_JOB_DESC=         P.AAC
                                                                  SLAVE_TASK_DESC=      P.AAE
                                                                        .EXTRN  PHN$_OK, PHN$_ANSWERED
                                                                        .EXTRN  PHN$_BUSYCALL, PHN$_CANCALL
                                                                        .EXTRN  PHN$_CANTREACH, PHN$_CONFCALL
                                                                        .EXTRN  PHN$_DEAD, PHN$_DECNETLINK
                                                                        .EXTRN  PHN$_DIRCAN, PHN$_FACSCAN
                                                                        .EXTRN  PHN$_HELPCAN, PHN$_HUNGUP
                                                                        .EXTRN  PHN$_JUSTRANG, PHN$_LOGGEDOFF
                                                                        .EXTRN  PHN$_REJECTED, PHN$_RING
                                                                        .EXTRN  PHN$_REJECTJUNK
                                                                        .EXTRN  PHN$_SENDINGMAIL
                                                                        .EXTRN  PHN$_BADCMD, PHN$_BADHELP
                                                                        .EXTRN  PHN$_BADMAILCMD
                                                                        .EXTRN  PHN$_BADSMB, PHN$_BADSPEC
                                                                        .EXTRN  PHN$_HELPMISSING
                                                                        .EXTRN  PHN$_IVREDUNANS
```

```
                                                               .EXTRN  PHN$_IVREDUNCALL
                                                               .EXTRN  PHN$_LINKERROR, PHN$_NEEDUSER
                                                               .EXTRN  PHN$_NOCALL, PHN$_NOHOLDS
                                                               .EXTRN  PHN$_NOPORTS, PHN$_NOPRIV
                                                               .EXTRN  PHN$_NOPROC, PHN$_NOTCONV
                                                               .EXTRN  PHN$_ONLYNODE, PHN$_PHONEBUSY
                                                               .EXTRN  PHN$_REMOTEERROR
                                                               .EXTRN  PHN$_TARGTERM, PHN$_UNPLUGGED
                                                               .EXTRN  PHN$_BADTERM, PHN$_SHAREDMBX
                                                               .EXTRN  PHN$_INPUTTERM, PHN$GQ_NODE_NAME
                                                               .EXTRN  PHN$GQ_SWITCH_HOOK
                                                               .EXTRN  PHN$GL_VIEWPORT_SIZE
                                                               .EXTRN  PHN$GB_SCROLL, PHN$GQ_PUBHEAD
                                                               .EXTRN  PHN$GB_FLAGS, PHN$TERM_CHARACTERISTIC
                                                               .EXTRN  PHN$CMP_TARGET, PHN$FRESH_SCREEN
                                                               .EXTRN  PHN$INFORM, PHN$KILL_PUB
                                                               .EXTRN  PHN$MAKE_PUB, PHN$MAKE_TSB
                                                               .EXTRN  PHN$QUEUE_SMB, PHN$READ_SLAVE
                                                               .EXTRN  UNS$NET_CONNECT

                                                               .PSECT  $CODE$,NOWRT,2

                                      0004 00000 MBX_AST:.WORD    Save R2
                   52      0000'    CF  9E 00002           MOVAB    MBX_IOSB, R2
                   5E               08  C2 00007           SUBL2    #8, SP
                   0A               62  E8 0000A           BLBS     MBX_IOSB, 1$
                   7E               62  3C 0000D           MOVZWL   MBX_IOSB, -(SP)
         00000000G 00               01  FB 00010           CALLS    #1, LIB$SIGNAL
     6E         02 A2               01  A3 00017 1$:       SUBW3    #1, MBX_IOSB+2, MBX_BUF_DSC
                04 AE      09       A2  9E 0001C           MOVAB    MBX_BUF+1, MBX_BUF_DSC+4
                                    5E  DD 00021           PUSHL    SP
                   7E      08       A2  9A 00023           MOVZBL   MBX_BUF, -(SP)
             0000G CF               02  FB 00027           CALLS    #2, PHN$QUEUE_SMB
             0000V CF               00  FB 0002C           CALLS    #0, PHN$MBX_ENABLE
                                        04 00031           RET

; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 0000
;  187          0512  2 ! To actually enable a read from our mailbox, we just do the appropriate
;  188          0513  2 ! mailbox $qio.
;  189          0514  2
;  190          0515  2 op = .phn$gg_pubhead[0];
;  191        P 0516  2 status = $qio(efn=phn$k_ourmbxefn,
;  192        P 0517  2               chan=.op[pub_w_channel],
;  193        P 0518  2               func=io$_readvblk,
;  194        P 0519  2               iosb=mbx_iosb,
;  195        P 0520  2               astadr=mbx_ast,
;  196        P 0521  2               p1=mbx_buf,
;  197          0522  2               p2=phn$k_mbxsize);
;  198          0523  2 check (.status);
;  199          0524  2 return;
;  200          0525  2
;  201          0526  1 end;


                                                         .EXTRN  SYS$QIO

                                    0000 00000           .ENTRY  PHN$MBX_ENABLE, Save nothing               ; 0475
                   50     0000G  CF   D0 00002           MOVL    PHN$GQ_PUBHEAD, OP                         ; 0515
                                 7E   7C 00007           CLRQ    -(SP)                                      ; 0522
                                 7E   7C 00009           CLRQ    -(SP)                                      ;
                   7E      0100  8F   3C 0000B           MOVZWL  #256, -(SP)                                ;
                          0000'  CF   9F 00010           PUSHAB  MBX_BUF                                    ;
                                 7E   D4 00014           CLRL    -(SP)                                      ;
                             B5  AF   9F 00016           PUSHAB  MBX_AST                                    ;
                          0000'  CF   9F 00019           PUSHAB  MBX_IOSB                                   ;
                                 31   DD 0001D           PUSHL   #49                                        ;
                   7E      00F4  C0   3C 0001F           MOVZWL  244(OP), -(SP)                             ;
                                 05   DD 00024           PUSHL   #5                                         ;
         00000000G 00            0C   FB 00026           CALLS   #12, SYS$QIO                               ;
                   09            50   E8 0002D           BLBS    STATUS, 1$                                 ; 0523
                                 50   DD 00030           PUSHL   STATUS                                     ;
         00000000G 00            01   FB 00032           CALLS   #1, LIB$SIGNAL                             ;
                                      04 00039 1$:       RET                                                ; 0526

; Routine Size:  58 bytes,    Routine Base:  $CODE$ + 0032
```

```
1 %sbttl 'PHN$ESTAB_LINK - Establish a Link'
1 !++
1 ! Functional Description:
1 !       This routine is called to establish a link between us and some other
1 !       place in the world.  There are four possible cases:
1 !               1.      A link between us and ourselves.
1 !               2.      A link between us and some other local user.
1 !               3.      A link between us and a remote node (for information).
1 !               4.      A link between us and some other remote user.
1 !
1 ! Formal Parameters:
1 !       target_spec     Address of descriptor of complete node/user name spec
1 !                       of the target.
1 !       pub_address     Address of longword in which to return address of PUB
1 !                       describing established link.  The PUB is marked
1 !                       temporary.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       phn$_badspec    Target spec syntax was invalid.
1 !       phn$_cantreach  Cannot reach the target right now.
1 !       phn$_needuser   Tried to establish link to home node w/o user name
1 !       phn$_nopriv     Do not have the necessary privileges.
1 !       phn$_noproc     No process owned by the user is available.
1 !       phn$_remoteerror Some sort of error during remote I/O.
1 !       phn$_targterm   None of the user's terminals are usable by PHONE.
1 !       DECnet status   Problem with remote link.
1 !
1 ! Side Effects:
1 !       Any error is displayed via PHN$INFORM; all callers should only be
1 !       checking the returned status.
1 !
1 !--
1
1
2 global routine phn$estab_link(target_spec,pub_address) = begin
2
2 own
2       path_error_done: byte;
2
3 routine path_error(error_code, fao_count, fao_text): novalue = begin
3     builtin
3       nullparameter;
4     if not .path_error_done then (
4       path_error_done = true;
4       if not nullparameter(3) then
4           phn$inform(.error_code, .fao_text)
4       else
4           phn$inform(.error_code);
3     );
2 end;
```

```
                                                                    .PSECT  $OWN$,NOEXE,2

                                                  00108 PATH_ERROR_DONE:
                                                                    .BLKB   1


                                                                    .PSECT  $CODE$,NOWRT,2

                                             0000 00000 PATH_ERROR:
                                                                    .WORD   Save nothing                             ; 0572
                          23      0000'  CF  E8 00002               BLBS    PATH_ERROR_DONE, 2$                       ; 0575
                  0000'   CF             01  90 00007               MOVB    #1, PATH_ERROR_DONE                      ; 0576
                          03             6C  91 0000C               CMPB    (AP), #3                                 ; 0577
                                         11  1F 0000F               BLSSU   1$                                       ;
                                    0C   AC  D5 00011               TSTL    12(AP)                                   ;
                                         0C  13 00014               BEQL    1$                                       ;
                                    0C   AC  DD 00016               PUSHL   FAO_TEXT                                 ; 0578
                                    04   AC  DD 00019               PUSHL   ERROR_CODE                               ;
                  0000G   CF             02  FB 0001C               CALLS   #2, PHN$INFORM                           ;
                                             04 00021               RET                                              ;
                                    04   AC  DD 00022 1$:           PUSHL   ERROR_CODE                               ; 0580
                  0000G   CF             01  FB 00025               CALLS   #1, PHN$INFORM                           ;
                                             04 0002A 2$:           RET                                              ; 0582

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 006C


;   259           0583  2
;   260           0584  2 bind
;   261           0585  2         target_spec_dsc = .target_spec: descriptor;
;   262           0586  2
;   263           0587  2 own
;   264           0588  2         own_described_buffer(user_name,12),
;   265           0589  2         parent_pid: long,
;   266           0590  2         own_described_buffer(term_number,7),
;   267           0591  2         path_list: block[dsc$k_d_bln, byte] field(descriptor_fields)
;   268           0592  2                 preset([dsc$w_length] = 0, [dsc$b_dtype] = dsc$k_dtype_t,
;   269           0593  2                         [dsc$b_class] = dsc$k_class_d, [dsc$a_pointer] = 0);
;   270           0594  2
;   271           0595  2 bind
;   272           0596  2         get_proc = uplit(word(12),word(jpi$_username),
;   273           0597  2                         long(user_name+8),
;   274           0598  2                         long(user_name),
;   275           0599  2                         word(4),word(jpi$_owner),
;   276           0600  2                         long(parent_pid),
;   277           0601  2                         long(0),
;   278           0602  2                         word(7),word(jpi$_terminal),
;   279           0603  2                         long(term_number+8),
;   280           0604  2                         long(term_number),
;   281           0605  2                         long(0));
;   282           0606  2
;   283           0607  2 local
;   284           0608  2         status: long,
;   285           0609  2         tp: ref pub,                    ! Pointer to new target PUB.
```

```
2               op: ref pub,                            ! Pointer to our PUB.
2               wild_pid: long,
2               potential: long, usable: long;
2
2       ! We begin by making a PUB for this link.  If we can't, just return a status.
2
2       status = phn$make_pub(.target_spec,.pub_address);
3       if .status eqlu phn$_badspec then (
3               phn$inform(phn$_badspec);
3               return phn$_badspec;
2       );
2       check (.status);
2       tp = ..pub_address;
2
2       ! Now we split up depending upon whether it is a local or remote link.
2
3       begin
3       bind
3               target_tsb = tp[pub_b_tsb]: tsb,
3               target_name = target_tsb[tsb_q_tkndsc,.target_tsb[tsb_w_tkncount]]: descriptor;
3
4       if not .target_tsb[tsb_v_remote] then (
4
4               ! We have a local link.  Make sure a user name was specified
4               ! in the spec.  We can't make a link just to our own node.
4
5               if not .target_tsb[tsb_v_user] then (
5                       phn$kill_pub(.tp);
5                       phn$inform(phn$_needuser);
5                       return phn$_needuser;
4               );
4
4               ! Now we split up depending upon whether we are linking to ourselves
4               ! or another local user.
4
4               op = .phn$gq_pubhead[0];
4               if phn$cmp_target(tp[pub_b_tsb],op[pub_b_tsb]) then
4
4                       ! We are linking to ourselves.  Just get the channel from
4                       ! our PUB and fill it into the new PUB.  If there isn't one
4                       ! we didn't have sufficient privilege.
4
5                       if .op[pub_w_channel] nequ 0 then (
5                               tp[pub_w_channel] = .op[pub_w_channel];
5                               return phn$_ok;
5                       ) else (
5                               phn$kill_pub(.tp);
5                               phn$inform(phn$_nopriv);
5                               return phn$_nopriv;
4                       );
```

```
        ! We are linking to another local user.  We must determine if
        ! anyone who is logged in fits the bill.

        potential = usable = 0;
        wild_pid = -1;
        loop (

                ! Get information on the next process.  If there aren't
                ! any more, then we're done.

                status = $getjpiw(efn=phn$k_getjpiefn,
                                  pidadr=wild_pid,
                                  itmlst=get_proc);

        exitif (.status eqlu ss$_nomoreproc);

                ! If we got a process, then determine if it is a detached
                ! interactive process owned by the target.

                if .status eqlu ss$_normal then (

                        if ch$eql(.target_name[len],.target_name[ptr], .user_name[len],.user_name[ptr],' ')
                           .parent_pid eqlu 0 and
                           .term_number[len] nequ 0 then (

                                ! We got a potential candidate.  Make sure that
                                ! their terminal is usable by PHONE.

                                inc (potential);
                                if phn$term_characteristic(term_number,tt$m_scope) then
                                        inc (usable);
                        );
                );
        );

        ! If there are no potential processes, or there are but no usable
        ! terminals, then return an appropriate status and flush the link.

        if .potential eqlu 0 then (
                phn$kill_pub(.tp);
                phn$inform(phn$_noproc);
                return phn$_noproc;
        );
        if .usable eqlu 0 then (
                phn$kill_pub(.tp);
                phn$inform(phn$_targterm);
                return phn$_targterm;
        );

        ! Now we must create a receive mailbox for the target.  We begin by
        ! building a name for the mailbox and trying to assign to it, in
        ! case someone else has already created it.

        begin
        local
                local_described_buffer(mbx_name,4+32);

```

```
	phn$mbx_name(target_name,mbx_name);
	status = $assign(devnam=mbx_name,
			 chan=tp[pub_w_channel]);
	if (.status nequ ss$_nosuchdev)
	    and (.status nequ ss$_ivdevnam) then (
		check (.status);
		return phn$_ok;
	);

	! Nope, mailbox doesn't already exist.  Create a permanent one with
	! the name and mark it for deletion so we don't leave crud around.

	status = $crembx(prmflg=1,
			 chan=tp[pub_w_channel],
			 maxmsg=phn$k_mbxsize,
			 lognam=mbx_name);
	if .status eqlu ss$_nopriv then (
		phn$kill_pub(.tp);
		phn$inform(phn$_nopriv);
		return phn$_nopriv;
	);
	check (.status);
	status = $delmbx(chan=.tp[pub_w_channel]);
	check (.status);
	return phn$_ok;

	end;
 );
```

```
  3 ! We are to establish a link to a remote node or user.  This requires
  3 ! us to make a logical link to the remote node.  The so-called task specifier
  3 ! is built as follows:
  3 !         normal case:      [node::...]node::"29="
  3 !         debugging:        {whatever is in PHN$SLAVE_TASK_SPECIFIER}
  3
  3 phn$inform(phn$_decnetlink);
  3
  4 begin
  4
  4 bind
  4         whole_target = target_tsb[tsb_q_tkndsc, 0]: descriptor;
  4
  4 local
  4         trnlnmlst : $itmlst_decl(items=1),
  4         local_described_buffer(specifier_buf, nam$c_maxrss);
  4
  4 !
  4 ! Translate PHN$SLAVE_TASK_SPECIFIER (used for debugging only).  Look in
  4 ! the process table and the job table (If not found in process table).
  4 !
P 4 $itmlst_init(itmlst=trnlnmlst,
P 4         (itmcod=lnm$_string,bufadr=.specifier_buf[ptr],
  4                 bufsiz=nam$c_maxrss,retlen=specifier_buf));
  4
P 4 status = $trnlnm(attr=%REF(lnm$m_case_blind),
P 4                 tabnam=lnm_process_desc,
P 4                 lognam=slave_task_desc,
  4                 itmlst=trnlnmlst);
  4
  4 if .status nequ ss$_normal
  5 then begin
  5     specifier_buf[len] = nam$c_maxrss;
P 5     status = $trnlnm(attr=%REF(lnm$m_case_blind),
P 5                 tabnam=lnm_job_desc,
P 5                 lognam=slave_task_desc,
  5                 itmlst=trnlnmlst);
  4     end;
  4
  5 if .status nequ ss$_normal then (
  5         ch$copy(.whole_target[len] - .target_name[len], .whole_target[ptr],
  5                 5, uplit byte('"29="'), ' ',
  5                 nam$c_maxrss, .specifier_buf[ptr]);
  5         specifier_buf[len] = .whole_target[len] - .target_name[len] + 5;
  4 );
  4
  4 ! Now we can actually create the logical link to the remote node.  This is
  4 ! using the UNS$NET_CONNECT routine.  It will use the "pass through" protocol
  4 ! if needed to make the connection.  The final routing is also returned.
  4
  4 path_error_done = false;
  4 status = uns$net_connect(specifier_buf, tp[pub_w_channel],
  4                         0, path_list, 0, path_error);
  5 if not .status then (
  5         phn$kill_pub(.tp);
  5         if not .path_error_done then
  5                 phn$inform(.status);
```

```
5               return .status;
4       );
4
4       ! Check to see if the routing needs updating.  If so, do so.
4
4       if not ch$eql(.path_list[len], .path_list[ptr],
4                       .whole_target[len] - .target_name[len], .whole_target[ptr],
5                       ' ') then (
5               local
5                       dsc_cnt, path_len, path_ptr;
5               ch$move(.target_name[len], .target_name[ptr], .specifier_buf[ptr]);
5               ch$move(.path_list[len], .path_list[ptr], .whole_target[ptr]);
5               ch$move(.target_name[len], .specifier_buf[ptr],
5                                               .whole_target[ptr] + .path_list[len]);
5               whole_target[len] = .path_list[len] + .target_name[len];
5               dsc_cnt = 1;
5               path_len = .whole_target[len];
5               path_ptr = .whole_target[ptr];
6               loop (
6                       bind
6                               dsc_ptr = target_tsb[tsb_q_tkndsc, .dsc_cnt]: descriptor;
7                       if (.path_len neq 0) and (ch$rchar(.path_ptr) eql '_') then (
7                               whole_target[len] = .whole_target[len] - 1;
7                               ch$move((path_len = .path_len - 1), .path_ptr + 1, .path_ptr);
6                       );
6                       dsc_ptr[len] = .path_len;
6                       dsc_ptr[ptr] = .path_ptr;
6                       exitif ((path_ptr = ch$find_ch(.path_len, .path_ptr, ':')) eql 0);
6                       path_ptr = .path_ptr + 2;
6                       dsc_ptr[len] = .path_ptr - .dsc_ptr[ptr];
6                       path_len = .path_len - .dsc_ptr[len];
6                       dsc_cnt = .dsc_cnt + 1;
5               );
5               target_tsb[tsb_w_tkncount] = .dsc_cnt;
4       );
4
3       end;
3
3       ! Now we have to do some more work if it's a link to a remote user.
3
4       if .target_tsb[tsb_v_user] then (
4
4               ! Now we send a special steering message to the network slave,
4               ! asking it to verify the existence of the target user.
4               ! It will send us back a status from that verification.
4
4               local
4                       local_described_buffer(verify_status,4);
4
4               phn$send_smb(.tp,smb__slave_verify,target_tsb[tsb_q_tkndsc,0]);
4               status = phn$read_slave(.tp[pub_w_channel],verify_status,true);
5               if .status nequ phn$_ok then (
5                       phn$break_link(.tp,smb__slave_done);
5                       phn$inform(.status);
5                       return .status;
4               );
5               if ..verify_status[ptr] nequ phn$_ok then (
;  537        0859  5                       phn$break_link(.tp,smb__slave_done);
;  538        0860  5                       phn$inform(..verify_status[ptr]);
;  539        0861  5                       return ..verify_status[ptr];
;  540        0862  4               );
;  541        0863  4
;  542        0864  3 );
;  543        0865  3
;  544        0866  3 phn$inform(0);
;  545        0867  3 return phn$_ok;
;  546        0868  2 end;
;  547        0869  2
;  548        0870  1 end;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                  000C  00044 P.AAG:    .WORD   12                                      ;
                                  0202  00046           .WORD   514                                     ;
                              00000000' 00048           .ADDRESS USER_NAME+8                            ;
                              00000000' 0004C           .ADDRESS USER_NAME                              ;
                                  0004  00050           .WORD   4                                       ;
                                  0303  00052           .WORD   771                                     ;
                              00000000' 00054           .ADDRESS PARENT_PID                             ;
                              00000000  00058           .LONG   0                                       ;
                                  0007  0005C           .WORD   7                                       ;
                                  031D  0005E           .WORD   797                                     ;
                              00000000' 00060           .ADDRESS TERM_NUMBER+8                          ;
                              00000000' 00064           .ADDRESS TERM_NUMBER                            ;
                              00000000  00068           .LONG   0                                       ;
                     22 3D 39 32 22     0006C P.AAH:    .ASCII  \"29="\                                 ;

                                                        .PSECT  $OWN$,NOEXE,2

                                        00109           .BLKB   3
                              0000000C  0010C USER_NAME:
                                                        .LONG   12                                      ;
                              00000000' 00110           .ADDRESS USER_NAME+8                            ;
                                        00114           .BLKB   12
                                        00120 PARENT_PID:
                                                        .BLKB   4
                              00000007  00124 TERM_NUMBER:
                                                        .LONG   7                                       ;
                              00000000' 00128           .ADDRESS TERM_NUMBER+8                          ;
                                        0012C           .BLKB   7
                                        00133           .BLKB   1
                                  0000  00134 PATH_LIST:
                                                        .WORD   0                                       ;
                                 02 0E  00136           .BYTE   14, 2                                   ;
                              00000000  00138           .LONG   0                                       ;

                                              GET_PROC=         P.AAG
                                                        .EXTRN  SYS$GETJPIW, SYS$ASSIGN
                                                        .EXTRN  SYS$CREMBX, SYS$DELMBX
                                                        .EXTRN  SYS$TRNLNM

                                                        .PSECT  $CODE$,NOWRT,2
```

```
                                                   0FFC 00000          .ENTRY  PHN$ESTAB_LINK, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0567
                                                                               R9,R10,R11                                    ;
                           5E        FEDC    CE  9E 00002              MOVAB   -292(SP), SP                                  ;
                           7E          04    AC  7D 00007              MOVQ    TARGET_SPEC, -(SP)                            ; 0616
                  0000G    CF                02  FB 0000B              CALLS   #2, PHN$MAKE_PUB                              ;
                           6E                50  D0 00010              MOVL    R0, STATUS                                    ;
              00000000G    8F                6E  D1 00013              CMPL    STATUS, #PHN$_BADSPEC                         ; 0617
                                             13  12 0001A              BNEQ    1$                                            ;
                                00000000G    8F  DD 0001C              PUSHL   #PHN$_BADSPEC                                 ; 0618
                  0000G    CF                01  FB 00022              CALLS   #1, PHN$INFORM                                ;
                           50   00000000G    8F  D0 00027              MOVL    #PHN$_BADSPEC, R0                             ; 0619
                                                 04 0002E              RET                                                   ;
                           09                6E  E8 0002F  1$:         BLBS    STATUS, 2$                                    ; 0621
                                             6E  DD 00032              PUSHL   STATUS                                        ;
              00000000G    00                01  FB 00034              CALLS   #1, LIB$SIGNAL                                ;
                           5A          08    BC  D0 0003B  2$:         MOVL    @PUB_ADDRESS, TP                              ; 0622
                           5B          0C    AA  9E 0003F              MOVAB   12(TP), R11                                   ; 0628
                           50          02    AB  3C 00043              MOVZWL  2(R11), R0                                    ; 0629
                           58          04  AB40  7E 00047              MOVAQ   4(R11)[R0], R8                                ;
                           03                6B  E9 0004C              BLBC    (R11), 3$                                     ; 0631
                                           017C  31 0004F              BRW     17$                                           ;
          1A               6B                01  E0 00052  3$:         BBS     #1, (R11), 4$                                 ; 0636
                                             5A  DD 00056              PUSHL   TP                                            ; 0637
                  0000G    CF                01  FB 00058              CALLS   #1, PHN$KILL_PUB                              ;
                                00000000G    8F  DD 0005D              PUSHL   #PHN$_NEEDUSER                                ; 0638
                  0000G    CF                01  FB 00063              CALLS   #1, PHN$INFORM                                ;
                           50   00000000G    8F  D0 00068              MOVL    #PHN$_NEEDUSER, R0                            ; 0639
                                                 04 0006F              RET                                                   ;
                           52        0000G   CF  D0 00070  4$:         MOVL    PHN$GQ_PUBHEAD, OP                            ; 0645
                                       0C    A2  9F 00075              PUSHAB  12(OP)                                        ; 0646
                                             5B  DD 00078              PUSHL   R11                                           ;
                  0000G    CF                02  FB 0007A              CALLS   #2, PHN$CMP_TARGET                            ;
                           12                50  E9 0007F              BLBC    R0, 6$                                        ;
                           50        00F4    C2  3C 00082              MOVZWL  244(OP), R0                                   ; 0652
                                             03  12 00087              BNEQ    5$                                            ;
                                           0100  31 00089              BRW     12$                                           ;
                   00F4    CA                50  B0 0008C  5$:         MOVW    R0, 244(TP)                                   ; 0653
                                           0314  31 00091              BRW     32$                                           ; 0655
                                             54  7C 00094  6$:         CLRQ    USABLE                                        ; 0663
                     08    AE                01  CE 00096              MNEGL   #1, WILD_PID                                  ; 0664
                                             7E  7C 0009A  7$:         CLRQ    -(SP)                                         ; 0672
                                             7E  D4 0009C              CLRL    -(SP)                                         ;
                                     0000'   CF  9F 0009E              PUSHAB  GET_PROC                                      ;
                                             7E  D4 000A2              CLRL    -(SP)                                         ;
                                       1C    AE  9F 000A4              PUSHAB  WILD_PID                                      ;
                                             01  DD 000A7              PUSHL   #1                                            ;
              00000000G    00                07  FB 000A9              CALLS   #7, SYS$GETJPIW                               ;
                           6E                50  D0 000B0              MOVL    R0, STATUS                                    ;
              000009A8     8F                6E  D1 000B3              CMPL    STATUS, #2472                                 ; 0674
                                             35  13 000BA              BEQL    8$                                            ;
                           01                6E  D1 000BC              CMPL    STATUS, #1                                    ; 0679
                                             D9  12 000BF              BNEQ    7$                                            ;
 0000'  CF           20    04    B8          68  2D 000C1              CMPC5   (R8), @4(R8), #32, USER_NAME, @USER_NAME+4    ; 0681
                                     0000'   DF     000C9                                                                    ;
                                             CC  12 000CC              BNEQ    7$                                            ;
                                     0000'   CF  D5 000CE              TSTL    PARENT_PID                                    ; 0682
```

```
                                                              I  4
LINKSUBS        LINKSUBS - Phone Link Subroutines             16-Sep-1984 02:11:44    VAX-11 Bliss-32 V4.0-742          Page  17
V04-000         PHN$ESTAB_LINK - Establish a Link             14-Sep-1984 12:53:27    [PHONE.SRC]LINKSUBS.B32;1              (7)

                                      C6   12 000D2           BNEQ    7$                                          ;
                             0000'    CF   B5 000D4           TSTW    TERM_NUMBER                                 ; 0683
                                      C0   13 000D8           BEQL    7$                                          ;
                                      55   D6 000DA           INCL    POTENTIAL                                   ; 0688
                     7E      1000     8F   3C 000DC           MOVZWL  #4096, -(SP)                                ; 0689
                             0000'    CF   9F 000E1           PUSHAB  TERM_NUMBER                                 ;
              0000G  CF               02   FB 000E5           CALLS   #2, PHN$TERM_CHARACTERISTIC                 ;
                     AD               50   E9 000EA           BLBC    R0, 7$                                      ;
                                      54   D6 000ED           INCL    USABLE                                      ; 0690
                                      A9   11 000EF           BRB     7$                                          ; 0664
                                      55   D5 000F1  8$:      TSTL    POTENTIAL                                   ; 0698
                                      1A   12 000F3           BNEQ    9$                                          ;
                                      5A   DD 000F5           PUSHL   TP                                          ; 0699
              0000G  CF               01   FB 000F7           CALLS   #1, PHN$KILL_PUB                            ;
                        00000000G     8F   DD 000FC           PUSHL   #PHN$_NOPROC                                ; 0700
              0000G  CF               01   FB 00102           CALLS   #1, PHN$INFORM                              ;
                     50 00000000G     8F   D0 00107           MOVL    #PHN$_NOPROC, R0                            ; 0701
                                           04 0010E           RET                                                 ;
                                      54   D5 0010F  9$:      TSTL    USABLE                                      ; 0703
                                      1A   12 00111           BNEQ    10$                                         ;
                                      5A   DD 00113           PUSHL   TP                                          ; 0704
              0000G  CF               01   FB 00115           CALLS   #1, PHN$KILL_PUB                            ;
                        00000000G     8F   DD 0011A           PUSHL   #PHN$_TARGTERM                              ; 0705
              0000G  CF               01   FB 00120           CALLS   #1, PHN$INFORM                              ;
                     50 00000000G     8F   D0 00125           MOVL    #PHN$_TARGTERM, R0                          ; 0706
                                           04 0012C           RET                                                 ;
                D4   AD               24   D0 0012D  10$:     MOVL    #36, MBX_NAME                               ; 0715
                D8   AD         DC    AD   9E 00131           MOVAB   MBX_NAME+8, MBX_NAME+4                      ;
                                D4    AD   9F 00136           PUSHAB  MBX_NAME                                    ; 0717
                                      58   DD 00139           PUSHL   R8                                          ;
              0000V  CF               02   FB 0013B           CALLS   #2, PHN$MBX_NAME                            ;
                                      7E   7C 00140           CLRQ    -(SP)                                       ; 0719
                     52       00F4    CA   9E 00142           MOVAB   244(TP), R2                                 ;
                                      52   DD 00147           PUSHL   R2                                          ;
                                D4    AD   9F 00149           PUSHAB  MBX_NAME                                    ;
          00000000G  00               04   FB 0014C           CALLS   #4, SYS$ASSIGN                              ;
                     6E               50   D0 00153           MOVL    R0, STATUS                                  ;
          00000908   8F               6E   D1 00156           CMPL    STATUS, #2312                               ; 0720
                                      0E   13 0015D           BEQL    11$                                         ;
          00000144   8F               6E   D1 0015F           CMPL    STATUS, #324                                ; 0721
                                      05   13 00166           BEQL    11$                                         ;
                     57               6E   E9 00168           BLBC    STATUS, 15$                                 ; 0722
                                      5E   11 0016B           BRB     16$                                         ; 0723
                                D4    AD   9F 0016D  11$:     PUSHAB  MBX_NAME                                    ; 0732
                                      7E   7C 00170           CLRQ    -(SP)                                       ;
                                      7E   D4 00172           CLRL    -(SP)                                       ;
                     7E      0100     8F   3C 00174           MOVZWL  #256, -(SP)                                 ;
                                      52   DD 00179           PUSHL   R2                                          ;
                                      01   DD 0017B           PUSHL   #1                                          ;
          00000000G  00               07   FB 0017D           CALLS   #7, SYS$CREMBX                              ;
                     6E               50   D0 00184           MOVL    R0, STATUS                                  ;
                     24               6E   D1 00187           CMPL    STATUS, #36                                 ; 0733
                                      1A   12 0018A           BNEQ    13$                                         ;
                                      5A   DD 0018C  12$:     PUSHL   TP                                          ; 0734
              0000G  CF               01   FB 0018E           CALLS   #1, PHN$KILL_PUB                            ;
                        00000000G     8F   DD 00193           PUSHL   #PHN$_NOPRIV                                ; 0735
              0000G  CF               01   FB 00199           CALLS   #1, PHN$INFORM                              ;
```

```
                                              J  4
LINKSUBS        LINKSUBS - Phone Link Subroutines           16-Sep-1984 02:11:44    VAX-11 Bliss-32 V4.0-742           Page  18
V04-000         PHN$ESTAB_LINK - Establish a Link           14-Sep-1984 12:53:27    [PHONE.SRC]LINKSUBS.B32;1                (7)

                                              50    00000000G   8F   D0 0019E        MOVL     #PHN$_NOPRIV, R0                      ; 0736
                                                                     04 001A5        RET
                                              09                6E   E8 001A6  13$:  BLBS     STATUS, 14$                           ; 0738
                                                                6E   DD 001A9        PUSHL    STATUS
                                    00000000G 00                01   FB 001AB        CALLS    #1, LIB$SIGNAL
                                              7E                62   3C 001B2  14$:  MOVZWL   (R2), -(SP)                           ; 0739
                                    00000000G 00                01   FB 001B5        CALLS    #1, SYS$DELMBX
                                              6E                50   D0 001BC        MOVL     R0, STATUS
                                              09                6E   E8 001BF        BLBS     STATUS, 16$                           ; 0740
                                                                6E   DD 001C2  15$:  PUSHL    STATUS
                                    00000000G 00                01   FB 001C4        CALLS    #1, LIB$SIGNAL
                                                          01DA       31 001CB  16$:  BRW      32$                                   ; 0741
                                                    00000000G   8F   DD 001CE  17$:  PUSHL    #PHN$_DECNETLINK                      ; 0751
                                        0000G CF                01   FB 001D4        CALLS    #1, PHN$INFORM
                                              56           04   AB   9E 001D9        MOVAB    4(R11), R6                            ; 0756
                                         0C   AE           FF   8F   9A 001DD        MOVZBL   #255, SPECIFIER_BUF                   ; 0760
                                         10   AE           14   AE   9E 001E2        MOVAB    SPECIFIER_BUF+8, SPECIFIER_BUF+4
                                              50           F0   AD   9E 001E7        MOVAB    TRNLNMLST, $$ITMBLKPTR                ; 0768
                                              80     000200FF   8F   D0 001EB        MOVL     #131327, ($$ITMBLKPTR)+
                                              80           10   AE   D0 001F2        MOVL     SPECIFIER_BUF+4, ($$ITMBLKPTR)+
                                              80           0C   AE   9E 001F6        MOVAB    SPECIFIER_BUF, ($$ITMBLKPTR)+
                                                                80   D4 001FA        CLRL     ($$ITMBLKPTR)+
                                                           F0   AD   9F 001FC        PUSHAB   TRNLNMLST                             ; 0773
                                                                7E   D4 001FF        CLRL     -(SP)
                                                        0000'   CF   9F 00201        PUSHAB   SLAVE_TASK_DESC
                                                        0000'   CF   9F 00205        PUSHAB   LNM_PROCESS_DESC
                                         14   AE     02000000   8F   D0 00209        MOVL     #33554432, 20(SP)
                                                           14   AE   9F 00211        PUSHAB   20(SP)
                                    00000000G 00                05   FB 00214        CALLS    #5, SYS$TRNLNM
                                              6E                50   D0 0021B        MOVL     R0, STATUS
                                              01                6E   D1 0021E        CMPL     STATUS, #1                            ; 0775
                                                                27   13 00221        BEQL     18$
                                         0C   AE           FF   8F   9B 00223        MOVZBW   #255, SPECIFIER_BUF                   ; 0777
                                                           F0   AD   9F 00228        PUSHAB   TRNLNMLST                             ; 0781
                                                                7E   D4 0022B        CLRL     -(SP)
                                                        0000'   CF   9F 0022D        PUSHAB   SLAVE_TASK_DESC
                                                        0000'   CF   9F 00231        PUSHAB   LNM_JOB_DESC
                                         14   AE     02000000   8F   D0 00235        MOVL     #33554432, 20(SP)
                                                           14   AE   9F 0023D        PUSHAB   20(SP)
                                    00000000G 00                05   FB 00240        CALLS    #5, SYS$TRNLNM
                                              6E                50   D0 00247        MOVL     R0, STATUS
                                              01                6E   D1 0024A  18$:  CMPL     STATUS, #1                            ; 0784
                                                                31   13 0024D        BEQL     20$
                                              59                66   3C 0024F        MOVZWL   (R6), R9                              ; 0785
                                              50                68   3C 00252        MOVZWL   (R8), R0
                                              59                50   C2 00255        SUBL2    R0, R9
                                         04   AE           FF   8F   9A 00258        MOVZBL   #255, 4(SP)
                                              57           10   AE   D0 0025D        MOVL     SPECIFIER_BUF+4, R7                   ; 0787
  04    AE              20               04   B6                59   2C 00261        MOVC5    R9, @4(R6), #32, 4(SP), (R7)
                                                                67      00268
                                                                10   18 00269        BGEQ     19$
                                              57                59   C0 0026B        ADDL2    R9, R7
                                         04   AE                59   C2 0026E        SUBL2    R9, 4(SP)
  04    AE              20            0000'   CF                05   2C 00272        MOVC5    #5, P.AAH, #32, 4(SP), (R7)
                                                                67      0027A
                    0C  AE                    59                05   A1 0027B  19$:  ADDW3    #5, R9, SPECIFIER_BUF                 ; 0788
                                                        0000'   CF   94 00280  20$:  CLRB     PATH_ERROR_DONE                       ; 0795
                                                    FD4D    CF   9F 00284          PUSHAB   PATH_ERROR                                    ; 0796
                                                            7E   D4 00288          CLRL     -(SP)                                         ;
                                                   0000'    CF   9F 0028A          PUSHAB   PATH_LIST                                     ;
                                                            7E   D4 0028E          CLRL     -(SP)                                         ;
                                                    00F4    CA   9F 00290          PUSHAB   244(TP)                                       ;
                                                      20    AE   9F 00294          PUSHAB   SPECIFIER_BUF                                 ;
                                  00000000G   00            06   FB 00297          CALLS    #6, UNS$NET_CONNECT                           ;
                                              6E            50   D0 0029E          MOVL     R0, STATUS                                    ;
                                              12            6E   E8 002A1          BLBS     STATUS, 22$                                   ; 0798
                                                            5A   DD 002A4          PUSHL    TP                                            ; 0799
                                      0000G   CF            01   FB 002A6          CALLS    #1, PHN$KILL_PUB                              ;
                                              03   0000'    CF   E9 002AB          BLBC     PATH_ERROR_DONE, 21$                          ; 0800
                                                          00CA   31 002B0          BRW      29$                                           ;
                                                          00C0   31 002B3 21$:     BRW      28$                                           ;
                                              57   0000'    CF   3C 002B6 22$:     MOVZWL   PATH_LIST, R7                                 ; 0807
                                              59            66   3C 002BB          MOVZWL   (R6), R9                                      ; 0808
                                              50            68   3C 002BE          MOVZWL   (R8), R0                                      ;
                                              59            50   C2 002C1          SUBL2    R0, R9                                        ;
            59               20       0000'   DF            57   2D 002C4          CMPC5    R7, @PATH_LIST+4, #32, R9, @4(R6)             ; 0807
                                                      04    B6      002CB                                                                 ;
                                                            6B   13 002CD          BEQL     27$                                           ;
                        10     BE          04   B8            68   28 002CF          MOVC3    (R8), @4(R8), @SPECIFIER_BUF+4                ; 0812
                        04     B6       0000'   DF            57   28 002D5          MOVC3    R7, @PATH_LIST+4, @4(R6)                      ; 0813
                        04   B647          10   BE            68   28 002DC          MOVC3    (R8), @SPECIFIER_BUF+4, @4(R6)[R7]            ; 0815
                             66               57            68   A1 002E3          ADDW3    (R8), R7, (R6)                                ; 0816
                                              59            01   D0 002E7          MOVL     #1, DSC_CNT                                   ; 0817
                                          04   AE            66   3C 002EA          MOVZWL   (R6), PATH_LEN                                ; 0818
                                              58      04    A6   D0 002EE          MOVL     4(R6), PATH_PTR                               ; 0819
                                              57      04  AB49   7E 002F2 23$:     MOVAQ    4(R11)[DSC_CNT], R7                           ; 0822
                                                      04    AE   D5 002F7          TSTL     PATH_LEN                                      ; 0823
                                                            11   13 002FA          BEQL     24$                                           ;
                                          5F   8F            68   91 002FC          CMPB     (PATH_PTR), #95                               ;
                                                            0B   12 00300          BNEQ     24$                                           ;
                                                            66   B7 00302          DECW     (R6)                                          ; 0824
                                                      04    AE   D7 00304          DECL     PATH_LEN                                      ; 0825
                             68          01   A8      04    AE   28 00307          MOVC3    PATH_LEN, 1(PATH_PTR), (PATH_PTR)             ;
                                              67      04    AE   B0 0030D 24$:     MOVW     PATH_LEN, (R7)                                ; 0827
                                          04   A7            58   D0 00311          MOVL     PATH_PTR, 4(R7)                               ; 0828
                             68          04   AE            3A   3A 00315          LOCC     #58, PATH_LEN, (PATH_PTR)                     ; 0829
                                                            02   12 0031A          BNEQ     25$                                           ;
                                                            51   D4 0031C          CLRL     R1                                            ;
                                              58            51   D0 0031E 25$:     MOVL     R1, PATH_PTR                                  ;
                                                            13   13 00321          BEQL     26$                                           ;
                                              58            02   C0 00323          ADDL2    #2, PATH_PTR                                  ; 0830
                             67               58      04    A7   A3 00326          SUBW3    4(R7), PATH_PTR, (R7)                         ; 0831
                                              50            67   3C 0032B          MOVZWL   (R7), R0                                      ; 0332
                                          04   AE            50   C2 0032E          SUBL2    R0, PATH_LEN                                  ;
                                                            59   D6 00332          INCL     DSC_CNT                                       ; 0833
                                                            BC   11 00334          BRB      23$                                           ; 0819
                                          02   AB            59   B0 00336 26$:     MOVW     DSC_CNT, 2(R11)                               ; 0835
                             63               6B            01   E1 0033A 27$:     BBC      #1, (R11), 31$                                ; 0842
                                          F4   AD            04   D0 0033E          MOVL     #4, VERIFY_STATUS                             ; 0849
                                          F8   AD      FC    AD   9E 00342          MOVAB    VERIFY_STATUS+8, VERIFY_STATUS+4              ;
                                                            56   DD 00347          PUSHL    R6                                            ; 0851
                                                            07   DD 00349          PUSHL    #7                                            ;
                                                            5A   DD 0034B          PUSHL    TP                                            ;
                                      0000V   CF            03   FB 0034D          CALLS    #3, PHN$SEND_SMB                              ;
```

```
                              01  DD 00352          PUSHL    #1                                   ; 0852
                      F4      AD  9F 00354          PUSHAB   VERIFY_STATUS                        ;
                7E    00F4    CA  3C 00357          MOVZWL   244(TP), -(SP)                       ;
        0000G   CF            03  FB 0035C          CALLS    #3, PHN$READ_SLAVE                   ;
                6E            50  D0 00361          MOVL     R0, STATUS                           ;
    00000000G   8F            6E  D1 00364          CMPL     STATUS, #PHN$_OK                     ; 0853
                              14  13 0036B          BEQL     30$                                  ;
                              0D  DD 0036D          PUSHL    #13                                  ; 0854
                              5A  DD 0036F          PUSHL    TP                                   ;
        0000V   CF            02  FB 00371          CALLS    #2, PHN$BREAK_LINK                   ;
                              6E  DD 00376 28$:     PUSHL    STATUS                               ; 0855
        0000G   CF            01  FB 00378          CALLS    #1, PHN$INFORM                       ;
                50            6E  D0 0037D 29$:     MOVL     STATUS, R0                           ; 0856
                              04     00380          RET                                           ;
    00000000G   8F      F8    BD  D1 00381 30$:     CMPL     @VERIFY_STATUS+4, #PHN$_OK           ; 0858
                              16  13 00389          BEQL     31$                                  ;
                              0D  DD 0038B          PUSHL    #13                                  ; 0859
                              5A  DD 0038D          PUSHL    TP                                   ;
        0000V   CF            02  FB 0038F          CALLS    #2, PHN$BREAK_LINK                   ;
                        F8    BD  DD 00394          PUSHL    @VERIFY_STATUS+4                     ; 0860
        0000G   CF            01  FB 00397          CALLS    #1, PHN$INFORM                       ;
                50      F8    BD  D0 0039C          MOVL     @VERIFY_STATUS+4, R0                 ; 0861
                              04     003A0          RET                                           ;
                              7E  D4 003A1 31$:     CLRL     -(SP)                                ; 0866
        0000G   CF            01  FB 003A3          CALLS    #1, PHN$INFORM                       ;
                50 00000000G  8F  D0 003A8 32$:     MOVL     #PHN$_OK, R0                         ; 0867
                              04     003AF          RET                                           ; 0870

; Routine Size:  944 bytes,    Routine Base:  $CODE$ + 0097
```

```
1 %sbttl 'PHN$MBX_NAME - Build Mailbox Name'
1 !++
1 ! Functional Description:
1 !       This routine is called to build a mailbox name for use in creating
1 !       a communication link mailbox.  The name consists of a user
1 !       name prefixed with "PHN$".
1 !
1 ! Formal Parameters:
1 !       user_name       Address of descriptor of user name.
1 !       name_buf        A descriptor for buffer to hold mailbox name.
1 !                       We set the length.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !       If a shared memory mailbox exists, a fatal error is signalled.
1 !
1 !--
1
1
2 global routine phn$mbx_name(user_name,name_buf): novalue = begin
2
2 bind
2       user_name_dsc = .user_name: descriptor,
2       name_buf_dsc = .name_buf: descriptor;
2
2 bind
2       name_table = ch$transtable(
2                       rep 36 of ('_'),
2                       '$',
2                       rep 11 of ('_'),
2                       '0','1','2','3','4','5','6','7','8','9',
2                       rep 7 of ('_'),
2                       'A','B','C','D','E','F','G','H','I','J','K','L','M',
2                       'N','O','P','Q','R','S','T','U','V','W','X','Y','Z',
2                       rep 6 of ('_'),
2                       'A','B','C','D','E','F','G','H','I','J','K','L','M',
2                       'N','O','P','Q','R','S','T','U','V','W','X','Y','Z',
2                       rep 5 of ('_'));
2
2 local
2       status: long,
2       local_described_buffer(shared_mailbox_name,4+4+16);
2
2
2 ! To build the mailbox name, we concatenate the prefix and the user name.
2 ! Then we run it through the translation table, converting any illegal
2 ! characters to underscores.
2
2 ch$copy(4,uplit byte('PHN$'), .user_name_dsc[len],.user_name_dsc[ptr],
                        ' ',.name_buf_dsc[len],.name_buf_dsc[ptr]);
        name_buf_dsc[len] = 4 + .user_name_dsc[len];
        ch$translate(name_table,.name_buf_dsc[len],.name_buf_dsc[ptr],' ',
                                        .name_buf_dsc[len],.name_buf_dsc[ptr]);

        ! Now build the name of the corresponding shared memory mailbox,
        ! MBX$PHN$username.  If a logical name exists by that name, then so
        ! does a mailbox.  We can't allow that, so signal a fatal error.

        begin
        local
                trnlnmlst : $itmlst_decl(items=1),
                local_described_buffer(result_buffer,nam$c_maxrss);

        ch$copy(4,uplit byte('MBX$'), .name_buf_dsc[len],.name_buf_dsc[ptr],
                ' ',.shared_mailbox_name[len],.shared_mailbox_name[ptr]);
        shared_mailbox_name[len] = 4 + .name_buf_dsc[len];

        $itmlst_init(itmlst=trnlnmlst,
                (itmcod=lnm$_string,bufadr=.result_buffer[ptr],
                        bufsiz=nam$c_maxrss,retlen=result_buffer));

        status = $trnlnm(attr=%ref(lnm$m_case_blind),
                        tabnam=$descriptor('LNM$SYSTEM'),
                        lognam=shared_mailbox_name,
                        acmode=%ref(psl$c_exec),
                        itmlst=trnlnmlst);
        if .status eqlu ss$_normal then
                signal(phn$_sharedmbx)
        else
            if .status nequ ss$_nolognam
                then check(.status);
        end;

        return;

        end;



                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                      00071             .BLKB   3
                  5F  00074 P.AAI:      .ASCII  \_\
                  5F  00075             .ASCII  \_\
                  5F  00076             .ASCII  \_\
                  5F  00077             .ASCII  \_\
                  5F  00078             .ASCII  \_\
                  5F  00079             .ASCII  \_\
                  5F  0007A             .ASCII  \_\
                  5F  0007B             .ASCII  \_\
                  5F  0007C             .ASCII  \_\
                  5F  0007D             .ASCII  \_\
                  5F  0007E             .ASCII  \_\
                  5F  0007F             .ASCII  \_\
                  5F  00080             .ASCII  \_\
                  5F  00081             .ASCII  \_\
5F  00082        .ASCII  \_\
5F  00083        .ASCII  \_\
5F  00084        .ASCII  \_\
5F  00085        .ASCII  \_\
5F  00086        .ASCII  \_\
5F  00087        .ASCII  \_\
5F  00088        .ASCII  \_\
5F  00089        .ASCII  \_\
5F  0008A        .ASCII  \_\
5F  0008B        .ASCII  \_\
5F  0008C        .ASCII  \_\
5F  0008D        .ASCII  \_\
5F  0008E        .ASCII  \_\
5F  0008F        .ASCII  \_\
5F  00090        .ASCII  \_\
5F  00091        .ASCII  \_\
5F  00092        .ASCII  \_\
5F  00093        .ASCII  \_\
5F  00094        .ASCII  \_\
5F  00095        .ASCII  \_\
5F  00096        .ASCII  \_\
5F  00097        .ASCII  \_\
24  00098        .ASCII  \$\
5F  00099        .ASCII  \_\
5F  0009A        .ASCII  \_\
5F  0009B        .ASCII  \_\
5F  0009C        .ASCII  \_\
5F  0009D        .ASCII  \_\
5F  0009E        .ASCII  \_\
5F  0009F        .ASCII  \_\
5F  000A0        .ASCII  \_\
5F  000A1        .ASCII  \_\
5F  000A2        .ASCII  \_\
5F  000A3        .ASCII  \_\
30  000A4        .ASCII  \0\
31  000A5        .ASCII  \1\
32  000A6        .ASCII  \2\
33  000A7        .ASCII  \3\
34  000A8        .ASCII  \4\
35  000A9        .ASCII  \5\
36  000AA        .ASCII  \6\
37  000AB        .ASCII  \7\
38  000AC        .ASCII  \8\
39  000AD        .ASCII  \9\
5F  000AE        .ASCII  \_\
5F  000AF        .ASCII  \_\
5F  000B0        .ASCII  \_\
5F  000B1        .ASCII  \_\
5F  000B2        .ASCII  \_\
5F  000B3        .ASCII  \_\
5F  000B4        .ASCII  \_\
41  000B5        .ASCII  \A\
42  000B6        .ASCII  \B\
43  000B7        .ASCII  \C\
44  000B8        .ASCII  \D\
45  000B9        .ASCII  \E\
46  000BA        .ASCII  \F\
47  000BB   .ASCII  \G\     ;
48  000BC   .ASCII  \H\     ;
49  000BD   .ASCII  \I\     ;
4A  000BE   .ASCII  \J\     ;
4B  000BF   .ASCII  \K\     ;
4C  000C0   .ASCII  \L\     ;
4D  000C1   .ASCII  \M\     ;
4E  000C2   .ASCII  \N\     ;
4F  000C3   .ASCII  \O\     ;
50  000C4   .ASCII  \P\     ;
51  000C5   .ASCII  \Q\     ;
52  000C6   .ASCII  \R\     ;
53  000C7   .ASCII  \S\     ;
54  000C8   .ASCII  \T\     ;
55  000C9   .ASCII  \U\     ;
56  000CA   .ASCII  \V\     ;
57  000CB   .ASCII  \W\     ;
58  000CC   .ASCII  \X\     ;
59  000CD   .ASCII  \Y\     ;
5A  000CE   .ASCII  \Z\     ;
5F  000CF   .ASCII  \_\     ;
5F  000D0   .ASCII  \_\     ;
5F  000D1   .ASCII  \_\     ;
5F  000D2   .ASCII  \_\     ;
5F  000D3   .ASCII  \_\     ;
5F  000D4   .ASCII  \_\     ;
41  000D5   .ASCII  \A\     ;
42  000D6   .ASCII  \B\     ;
43  000D7   .ASCII  \C\     ;
44  000D8   .ASCII  \D\     ;
45  000D9   .ASCII  \E\     ;
46  000DA   .ASCII  \F\     ;
47  000DB   .ASCII  \G\     ;
48  000DC   .ASCII  \H\     ;
49  000DD   .ASCII  \I\     ;
4A  000DE   .ASCII  \J\     ;
4B  000DF   .ASCII  \K\     ;
4C  000E0   .ASCII  \L\     ;
4D  000E1   .ASCII  \M\     ;
4E  000E2   .ASCII  \N\     ;
4F  000E3   .ASCII  \O\     ;
50  000E4   .ASCII  \P\     ;
51  000E5   .ASCII  \Q\     ;
52  000E6   .ASCII  \R\     ;
53  000E7   .ASCII  \S\     ;
54  000E8   .ASCII  \T\     ;
55  000E9   .ASCII  \U\     ;
56  000EA   .ASCII  \V\     ;
57  000EB   .ASCII  \W\     ;
58  000EC   .ASCII  \X\     ;
59  000ED   .ASCII  \Y\     ;
5A  000EE   .ASCII  \Z\     ;
5F  000EF   .ASCII  \_\     ;
5F  000F0   .ASCII  \_\     ;
5F  000F1   .ASCII  \_\     ;
5F  000F2   .ASCII  \_\     ;
5F  000F3   .ASCII  \_\     ;
```

```
                                    24  4E  48  50  000F4 P.AAJ:  .ASCII   \PHN$\                                        ;
                                    24  58  42  4D  000F8 P.AAK:  .ASCII   \MBX$\                                        ;
                4D  45  54  53  59  53  24  4D  4E  4C  000FC P.AAM:  .ASCII   \LNM$SYSTEM\                                  ;
                                                    00106         .BLKB    2
                                         0000000A   00108 P.AAL:  .LONG    10                                            ;
                                         00000000'  0010C         .ADDRESS P.AAM                                         ;

                                                          NAME_TABLE=        P.AAI


                                                                  .PSECT   $CODE$,NOWRT,2

                                                07FC 00000        .ENTRY   PHN$MBX_NAME, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0898
                                                                           R10                                           ;
                              5E       FEC0   CE  9E 00002        MOVAB    -320(SP), SP                                  ;
                              57         04   AC  D0 00007        MOVL     USER_NAME, R7                                 ; 0901
                              56         08   AC  D0 0000B        MOVL     NAME_BUF, R6                                  ; 0902
                         E0   AD              18  D0 0000F        MOVL     #24, SHARED_MAILBOX_NAME                      ; 0920
                         E4   AD         E8   AD  9E 00013        MOVAB    SHARED_MAILBOX_NAME+8, -                      ;
                                                                           SHARED_MAILBOX_NAME+4                         ;
                              5A              67  3C 00018        MOVZWL   (R7), R10                                     ; 0927
                              59              66  3C 0001B        MOVZWL   (R6), R9                                      ; 0928
                              58         04   A6  D0 0001E        MOVL     4(R6), R8                                     ;
        59          20   0000'   CF              04  2C 00022        MOVC5    #4, P.AAJ, #32, R9, (R8)                      ;
                                              68     00029                                                               ;
                                              0D  18 0002A        BGEQ     1$                                            ;
                              58              04  C0 0002C        ADDL2    #4, R8                                        ;
                              59              04  C2 0002F        SUBL2    #4, R9                                        ;
        59          20     04    B7              5A  2C 00032        MOVC5    R10, @4(R7), #32, R9, (R8)                    ;
                                              68     00038                                                               ;
                    66           67              04  A1 00039 1$:    ADDW3    #4, (R7), (R6)                                ; 0929
0000'   CF          20     04    B6              66  2E 0003D        MOVTC    (R6), @4(R6), #32, NAME_TABLE, (R6), @4(R6) ; 0931
                           04    B6              66     00045                                                            ;
                           08    AE         FF   8F  9A 00048        MOVZBL   #255, RESULT_BUFFER                           ; 0940
                           0C    AE         10   AE  9E 0004D        MOVAB    RESULT_BUFFER+8, RESULT_BUFFER+4              ;
                                 59              66  3C 00052        MOVZWL   (R6), R9                                      ; 0942
                                 58         E0   AD  3C 00055        MOVZWL   SHARED_MAILBOX_NAME, R8                       ; 0943
                                 57         E4   AD  D0 00059        MOVL     SHARED_MAILBOX_NAME+4, R7                     ;
        58          20   0000'   CF              04  2C 0005D        MOVC5    #4, P.AAK, #32, R8, (R7)                      ;
                                                 67     00064                                                            ;
                                                 0D  18 00065        BGEQ     2$                                            ;
                                 57              04  C0 00067        ADDL2    #4, R7                                        ;
                                 58              04  C2 0006A        SUBL2    #4, R8                                        ;
        58          20     04    B6              59  2C 0006D        MOVC5    R9, @4(R6), #32, R8, (R7)                     ;
                                                 67     00073                                                            ;
              E0    AD           66              04  A1 00074 2$:    ADDW3    #4, (R6), SHARED_MAILBOX_NAME                 ; 0944
                                 50         D0   AD  9E 00079        MOVAB    TRNLNMLST, $$ITMBLKPTR                        ; 0948
                                 80   000200FF   8F  D0 0007D        MOVL     #131327, ($$ITMBLKPTR)+                       ;
                                 80         0C   AE  D0 00084        MOVL     RESULT_BUFFER+4, ($$ITMBLKPTR)+               ;
                                 80         08   AE  9E 00088        MOVAB    RESULT_BUFFER, ($$ITMBLKPTR)+                 ;
                                                 80  D4 0008C        CLRL     ($$ITMBLKPTR)+                                ;
                                            D0   AD  9F 0008E        PUSHAB   TRNLNMLST                                     ; 0954
                           08    AE              01  D0 00091        MOVL     #1, 8(SP)                                     ;
                                            08   AE  9F 00095        PUSHAB   8(SP)                                         ;
                                            E0   AD  9F 00098        PUSHAB   SHARED_MAILBOX_NAME                           ;
                                         0000'   CF  9F 0009B        PUSHAB   P.AAL                                         ;
                           10    AE   02000000   8F  D0 0009F        MOVL     #33554432, 16(SP)                             ;
```

```
                                      10    AE  9F 000A7          PUSHAB  16(SP)                                          ;
                      00000000G  00         05  FB 000AA          CALLS   #5, SYS$TRNLNM                                  ;
                                 52         50  D0 000B1          MOVL    R0, STATUS                                      ;
                                 01         52  D1 000B4          CMPL    STATUS, #1                                      ; 0955
                                            08  12 000B7          BNEQ    3$                                              ;
                                 00000000G  8F  DD 000B9          PUSHL   #PHN$_SHAREDMBX                                 ; 0956
                                            0E  11 000BF          BRB     4$                                              ;
                      000001BC   8F         52  D1 000C1 3$:      CMPL    STATUS, #444                                    ; 0958
                                            0C  13 000C8          BEQL    5$                                              ;
                                 09         52  E8 000CA          BLBS    STATUS, 5$                                      ; 0959
                                            52  DD 000CD          PUSHL   STATUS                                          ;
                      00000000G  00         01  FB 000CF 4$:      CALLS   #1, LIB$SIGNAL                                  ;
                                                04 000D6 5$:      RET                                                     ; 0964

; Routine Size:  215 bytes,    Routine Base:  $CODE$ + 0447
```

```
%sbttl 'PHN$BREAK_LINK - Break a Link'
!++
! Functional Description:
!       This routine is called to break a link between us and some other
!       person or node.
!
! Formal Parameters:
!       target_pub      The address of the PUB describing the link.
!       smb_type        The type code of the steering message to be sent
!                       as a reason for breaking the link.
!       smb_msg         An optional message text for the steering message.
!
! Implicit Inputs:
!       global data
!
! Implicit Outputs:
!       global data
!
! Returned Value:
!       none
!
! Side Effects:
!
!--


global routine phn$break_link(target_pub,smb_type,smb_msg): novalue = begin

bind
        tp = .target_pub: pub,
        target_tsb = tp[pub_b_tsb]: tsb;

local
        status: long;

builtin
        argptr;


! First we send a message with the reason for breaking the link.

callg(argptr(),phn$send_smb);

! If this is a remote link, we have to send the slave a special steering
! message to tell it to go away.  If this message wasn't sent above, then
! do it now.  Then we can clear away the network logical link.

if .target_tsb[tsb_v_remote] then (
        if .smb_type nequ smb__slave_done then
                phn$send_smb(tp,smb__slave_done);
        status = $dassgn(chan=.tp[pub_w_channel]);
        check (.status);
);

! Finally we can kill the PUB representing the link we have broken.

phn$kill_pub(tp);
```

```
;  702          1022  2 return;
;  703          1023  2
;  704          1024  1 end;


                                                                      .EXTRN  SYS$DASSGN

                                              0004 00000              .ENTRY  PHN$BREAK_LINK, Save R2                      ; 0991
                             52         04  AC  D0 00002              MOVL    TARGET_PUB, R2                               ; 0994
                     0000V   CF             6C  FA 00006              CALLG   (AP), PHN$SEND_SMB                           ; 1006
                             27         0C  A2  E9 0000B              BLBC    12(R2), 2$                                   ; 1012
                             0D         08  AC  D1 0000F              CMPL    SMB_TYPE, #13                                ; 1013
                                            09  13 00013              BEQL    1$                                           ;
                                            0D  DD 00015              PUSHL   #13                                          ; 1014
                                            52  DD 00017              PUSHL   R2                                           ;
                     0000V   CF             02  FB 00019              CALLS   #2, PHN$SEND_SMB                             ;
                             7E       00F4  C2  3C 0001E 1$:          MOVZWL  244(R2), -(SP)                               ; 1015
                 00000000G   00             01  FB 00023              CALLS   #1, SYS$DASSGN                               ;
                             09             50  E8 0002A              BLBS    STATUS, 2$                                   ; 1016
                                            50  DD 0002D              PUSHL   STATUS                                       ;
                 00000000G   00             01  FB 0002F              CALLS   #1, LIB$SIGNAL                               ;
                                            52  DD 00036 2$:          PUSHL   R2                                           ; 1021
                     0000G   CF             01  FB 00038              CALLS   #1, PHN$KILL_PUB                             ;
                                                04 0003D              RET                                                  ; 1024

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 051E
;  706         1025  1 %sbttl 'PHN$BREAK_CALL - Break Link to Person We Are Calling'
;  707         1026  1 !++
;  708         1027  1 ! Functional Description:
;  709         1028  1 !       This routine is called to break the link to someone we are
;  710         1029  1 !       currently trying to call.  It was made a seperate routine because
;  711         1030  1 !       it is done in a lot of places.
;  712         1031  1 !
;  713         1032  1 ! Formal Parameters:
;  714         1033  1 !       none
;  715         1034  1 !
;  716         1035  1 ! Implicit Inputs:
;  717         1036  1 !       global data
;  718         1037  1 !
;  719         1038  1 ! Implicit Outputs:
;  720         1039  1 !       global data
;  721         1040  1 !
;  722         1041  1 ! Returned Value:
;  723         1042  1 !       none
;  724         1043  1 !
;  725         1044  1 ! Side Effects:
;  726         1045  1 !
;  727         1046  1 !--
;  728         1047  1
;  729         1048  1
;  730         1049  2 global routine phn$break_call: novalue = begin
;  731         1050  2
;  732         1051  2 local
;  733         1052  2         op: ref pub;                    ! Pointer to our PUB.
;  734         1053  2
;  735         1054  2
;  736         1055  2 ! We have to clear the calling flag and busylink in our PUB because we
;  737         1056  2 ! will no longer be calling the person.  PHN$BREAK_LINK is then called to
;  738         1057  2 ! do its usual link-breaking stuff.  Finally, we better cancel any outstanding
;  739         1058  2 ! timer requests, in case there is one waiting to ring the person's phone.
;  740         1059  2
;  741         1060  2 op = .phn$gq_pubhead[0];
;  742         1061  2 op[pub_v_calling] = false;
;  743         1062  2 phn$break_link(.op[pub_l_busylink],smb__hungup);
;  744         1063  2 op[pub_l_busylink] = 0;
;  745         1064  2 $cantim();
;  746         1065  2 return;
;  747         1066  2
;  748         1067  1 end;


                                                            .EXTRN  SYS$CANTIM

                                         0004 00000         .ENTRY  PHN$BREAK_CALL, Save R2                         ; 1049
                  52      0000G   CF  D0 00002              MOVL    PHN$GQ_PUBHEAD, OP                              ; 1060
          00F0    C2              08  8A 00007              BICB2   #8, 240(OP)                                     ; 1061
                                  09  DD 0000C              PUSHL   #9                                              ; 1062
                          00F8    C2  DD 0000E              PUSHL   248(OP)                                         ;
            AC    AF              02  FB 00012              CALLS   #2, PHN$BREAK_LINK                              ;
                          00F8    C2  D4 00016              CLRL    248(OP)                                         ; 1063
                                  7E  7C 0001A              CLRQ    -(SP)                                           ; 1064
     00000000G    00              02  FB 0001C              CALLS   #2, SYS$CANTIM                                  ;
```

```
                                              04 00023          RET                                              ; 1067

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 055C
```

```
1 %sbttl 'PHN$SEND_SMB - Send Steering Message'
1 !++
1 ! Functional Description:
1 !       This routine is called to send a steering message to another person
1 !       or node represented by a PUB.  The message that is sent consists of
1 !       up to three parts, as follows:
1 !
1 !               1.      The 1-byte steering message code.
1 !
1 !               2.      Our own node string, representing the sender of
1 !                       the message, followed by an eofrom character.
1 !
1 !               3.      The optional message text.
1 !
1 ! Formal Parameters:
1 !       target_pub      Address of the PUB representing the destination.
1 !       smb_type        The steering message type code.
1 !       smb_msg         Address of descriptor of optional message text.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine phn$send_smb(target_pub,smb_type,smb_msg): novalue = begin
2
2 bind
2       tp = .target_pub: pub,
2       target_tsb = tp[pub_b_tsb]: tsb,
2       smb_msg_dsc = .smb_msg: descriptor;
2
2 local
2       status: long,
2       op: ref pub,                            ! Pointer to our PUB.
2       msg_buf: vector[phn$k_mbxsize,byte],
2       buf_i: long,
2       iosb: block[8,byte];
2
2 builtin
2       nullparameter;
2
2
2 ! First we have to build the message that we are going to send.  Begin
2 ! with the message type code.
2
2 buf_i = 0;
2
2 msg_buf[.buf_i] = .smb_type<0,8,0>;
```

```
  2 inc (buf_i);
  2
  2 ! Now include our spec, representing the sender of the message.
  2
  2 op = .phn$gq_pubhead[0];
  3 begin
  3 bind
  3         our_tsb = op[pub_b_tsb]: tsb,
  3         our_spec_dsc = our_tsb[tsb_q_tkndsc,0]: descriptor;
  3
  3 ! If the token count is above 2 we have a "pass through" connection so
  3 ! develope the reverse routing back to us.  But, if we are a slave task then
  3 ! the routing has already been performed.  In which case we don't do anything
  3 ! because we have arrived at the desired local location.  Being the slave task
  3 ! is when the OP pub has the total routing contained.  Any other time it would
  3 ! be our own home node specification.
  3
  3 if (.target_tsb[tsb_w_tkncount] gtru 2) and
  4    (.our_tsb[tsb_w_tkncount] lequ 2) then (
  4     decr node_index from .target_tsb[tsb_w_tkncount] - 2 to 1 do
  5         begin
  5         bind
  5             node_dsc = target_tsb[tsb_q_tkndsc, .node_index]: descriptor;
  5         ch$move(.node_dsc[len], .node_dsc[ptr], msg_buf[.buf_i]);
  5         buf_i = .buf_i + .node_dsc[len];
  4         end;
  3 );
  3
  3 ch$move(.our_spec_dsc[len],.our_spec_dsc[ptr],msg_buf[.buf_i]);
  3 buf_i = .buf_i + .our_spec_dsc[len];
  3 msg_buf[.buf_i] = eofrom;
  3 inc (buf_i);
  2 end;
  2
  2 ! Finally, if there is optional message text, we have to move it into
  2 ! the buffer.  Make sure we don't go off the end of the buffer.
  2
  3 if not nullparameter(3) then (
  3         local
  3                 length: long;
  3
  3         length = minu(.smb_msg_dsc[len],phn$k_mbxsize-.buf_i);
  3         ch$move(.length,.smb_msg_dsc[ptr],msg_buf[.buf_i]);
  3         buf_i = .buf_i + .length;
  2 );
  2
  2 ! Now we split up depending upon whether it's a local or remote message.
  2 ! If it's local, we can just send the message to the user's receive
  2 ! mailbox.  If we get an error doing so, tell the user.
  2
  3 if not .target_tsb[tsb_v_remote] then (
P 3         status = $qiow(chan=.tp[pub_w_channel],
P 3                        func=io$_writevblk + io$m_now,
P 3                        iosb=iosb,
P 3                        p1=msg_buf,
  3                        p2=.buf_i);
  3          if .status nequ ss$_normal or .iosb[0,0,16,0] nequ ss$_normal then
;   864           1182  3                   phn$inform(phn$_linkerror);
;   865           1183  3 ) else (
;   866           1184  3
;   867           1185  3           ! It is a remote send.  All we have to do is send the message over the
;   868           1186  3           ! logical link.  If we get an error, tell the user.
;   869           1187  3
;   870         P 1188  3           status = $qiow(efn=phn$k_decnetefn,
;   871         P 1189  3                          chan=.tp[pub_w_channel],
;   872         P 1190  3                          func=io$_writevblk,
;   873         P 1191  3                          iosb=iosb,
;   874         P 1192  3                          p1=msg_buf,
;   875           1193  3                          p2=.buf_i);
;   876           1194  3           if .status nequ ss$_normal or .iosb[0,0,16,0] nequ ss$_normal then
;   877           1195  3                   phn$inform(phn$_linkerror);
;   878           1196  2 );
;   879           1197  2
;   880           1198  2 return;
;   881           1199  2
;   882           1200  1 end;



                                                                 .EXTRN   SYS$QIOW

                                           OFFC 00000            .ENTRY   PHN$SEND_SMB, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  1101
                                                                          R10,R11                                      ;
                        5E       FEF8   CE   9E 00002            MOVAB    -264(SP), SP                                 ;
                        5B         04   AC   D0 00007            MOVL     TARGET_PUB, R11                              ;  1104
                        5A         0C   AB   9E 0000B            MOVAB    12(R11), R10                                 ;  1105
                                   0C   AC   DD 0000F            PUSHL    SMB_MSG                                      ;  1106
                                        56   D4 00012            CLRL     BUF_I                                        ;  1122
                   0C AE46         08   AC   90 00014            MOVB     SMB_TYPE, MSG_BUF[BUF_I]                     ;  1124
                                        56   D6 0001A            INCL     BUF_I                                        ;  1125
                        50      0000G   CF   D0 0001C            MOVL     PHN$GQ_PUBHEAD, R0                           ;  1129
                        50              0C   C0 00021            ADDL2    #12, R0                                      ;  1132
                        59         04   A0   9E 00024            MOVAB    4(R0), R9                                    ;  1133
                        02         02   AA   B1 00028            CMPW     2(R10), #2                                   ;  1142
                                        23   1B 0002C            BLEQU    3$                                           ;
                        02         02   A0   B1 0002F            CMPW     2(R0), #2                                    ;  1143
                                        1D   1A 00032            BGTRU    3$                                           ;
                        58         02   AA   3C 00034            MOVZWL   2(R10), NODE_INDEX                           ;  1144
                                        58   D7 00038            DECL     NODE_INDEX                                   ;
                                        12   11 0003A            BRB      2$                                           ;
                        57      04 AA48   7E 0003C  1$:          MOVAQ    4(R10)[NODE_INDEX], R7                       ;  1147
    0C AE46        04   B7              67   28 00041            MOVC3    (R7), @4(R7), MSG_BUF[BUF_I]                 ;  1148
                        50              67   3C 00048            MOVZWL   (R7), R0                                     ;  1149
                        56              50   C0 0004B            ADDL2    R0, BUF_I                                    ;
                        EB              58   F5 0004E  2$:       SOBGTR   NODE_INDEX, 1$                               ;  1144
    0C AE46        04   B9              69   28 00051  3$:       MOVC3    (R9), @4(R9), MSG_BUF[BUF_I]                 ;  1153
                        50              69   3C 00058            MOVZWL   (R9), R0                                     ;  1154
                        56              50   C0 0005B            ADDL2    R0, BUF_I                                    ;
                                   0C AE46   94 0005E            CLRB     MSG_BUF[BUF_I]                               ;  1155
                                        56   D6 00062            INCL     BUF_I                                        ;  1156
                        03              6C   91 00064            CMPB     (AP), #3                                     ;  1162
                                        29   1F 00067            BLSSU    5$                                           ;
                                   0C   AC   D5 00069            TSTL     12(AP)                                       ;
                                        24   13 0006C            BEQL     5$                                           ;
```

```
                    51 00000100   8F           56 C3 0006E        SUBL3   BUF_I, #256, R1                    ; 1166
                                  50      00   BE 3C 00076        MOVZWL  @0(SP), R0                         ;
                                  51           50 D1 0007A        CMPL    R0, R1                             ;
                                               03 1B 0007D        BLEQU   4$                                 ;
                                  50           51 D0 0007F        MOVL    R1, R0                             ;
                                  58           50 D0 00082 4$:    MOVL    R0, LENGTH                         ;
                         57       6E           04 C1 00085        ADDL3   #4, (SP), R7                       ; 1167
                    0C AE46       97           58 28 00089        MOVC3   LENGTH, @(R7)+, MSG_BUF[BUF_I]     ;
                                  56           58 C0 0008F        ADDL2   LENGTH, BUF_I                      ; 1168
                                  27           6A E8 00092 5$:    BLBS    (R10), 6$                          ; 1175
                                               7E 7C 00095        CLRQ    -(SP)                              ; 1180
                                               7E 7C 00097        CLRQ    -(SP)                              ;
                                               56 DD 00099        PUSHL   BUF_I                              ;
                                          20   AE 9F 0009B        PUSHAB  MSG_BUF                            ;
                                               7E 7C 0009E        CLRQ    -(SP)                              ;
                                          24   AE 9F 000A0        PUSHAB  IOSB                               ;
                                  7E      70   8F 9A 000A3        MOVZBL  #112, -(SP)                        ;
                                  7E    00F4   CB 3C 000A7        MOVZWL  244(R11), -(SP)                    ;
                                               7E D4 000AC        CLRL    -(SP)                              ;
                          00000000G 00         0C FB 000AE        CALLS   #12, SYS$QIOW                      ;
                                  01           50 D1 000B5        CMPL    STATUS, #1                         ; 1181
                                               25 13 000B8        BEQL    7$                                 ;
                                               29 11 000BA        BRB     8$                                 ; 1182
                                               7E 7C 000BC 6$:    CLRQ    -(SP)                              ; 1193
                                               7E 7C 000BE        CLRQ    -(SP)                              ;
                                               56 DD 000C0        PUSHL   BUF_I                              ;
                                          20   AE 9F 000C2        PUSHAB  MSG_BUF                            ;
                                               7E 7C 000C5        CLRQ    -(SP)                              ;
                                          24   AE 9F 000C7        PUSHAB  IOSB                               ;
                                               30 DD 000CA        PUSHL   #48                                ;
                                  7E    00F4   CB 3C 000CC        MOVZWL  244(R11), -(SP)                    ;
                                               04 DD 000D1        PUSHL   #4                                 ;
                          00000000G 00         0C FB 000D3        CALLS   #12, SYS$QIOW                      ;
                                  01           50 D1 000DA        CMPL    STATUS, #1                         ; 1194
                                               06 12 000DD        BNEQ    8$                                 ;
                                  01      04   AE B1 000DF 7$:    CMPW    IOSB, #1                           ;
                                               0B 13 000E3        BEQL    9$                                 ;
                                      00000000G 8F DD 000E5 8$:   PUSHL   #PHN$_LINKERROR                    ; 1195
                              0000G CF         01 FB 000EB        CALLS   #1, PHN$INFORM                     ;
                                                  04 000F0 9$:    RET                                        ; 1200

; Routine Size:  241 bytes,    Routine Base:  $CODE$ + 0580
```

```
1 %sbttl 'PHN$FORCE_LINKS - Force Links to New Person'
1 !++
1 ! Functional Description:
1 !       This routine is called when a new person enters the conversation
1 !       as a result of answering our call.  We need to inform everyone
1 !       else in the conversation about the new person, and also inform
1 !       the new person about them.  This routine is the principle vehicle
1 !       for setting up conference calls.
1 !
1 ! Formal Parameters:
1 !       new_pub         Address of new person's PUB.
1 !
1 ! Implicit Inputs:
1 !       global data
1 !
1 ! Implicit Outputs:
1 !       global data
1 !
1 ! Returned Value:
1 !       none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine phn$force_links(new_pub): novalue = begin
2
2 bind
2       np = .new_pub: pub,
2       new_tsb = np[pub_b_tsb]: tsb;
2
2 local
2       p: ref pub;
2
2
2 ! We scan the PUB chain (but not our own), looking for people in the
2 ! current conversation.  Make sure not to find the new person's PUB.
2
2 p = ..phn$gq_pubhead[0];
3 until .p eqla phn$gq_pubhead do (
3       if not .p[pub_v_temporary] and
3          not .p[pub_v_uhaveheld] and
4          (.p nega np)                 then (
4
4               ! We found a PUB.  Tell this person about the new guy.
4               ! Tell the new guy about this person.
4
4               bind
4                       person_tsb = p[pub_b_tsb]: tsb;
4
4               phn$send_smb(.p,smb__forced_link,new_tsb   [tsb_q_tkndsc,0]);
4               phn$send_smb(np,smb__forced_link,person_tsb[tsb_q_tkndsc,0]);
3       );
3
3       p = .p[pub_l_flink];
2 );
```

```
;  941          1258  2
;  942          1259  2 return;
;  943          1260  2
;  944          1261  1 end;



                                                    000C 00000           .ENTRY   PHN$FORCE_LINKS, Save R2,R3                  ; 1227
                    53      04    AC             0C   C1 00002           ADDL3    #12, NEW_PUB, R3                             ; 1231
                                  52     0000G  DF   D0 00007           MOVL     @PHN$GQ_PUBHEAD, P                           ; 1240
                                  50     0000G  CF   9E 0000C 1$:       MOVAB    PHN$GQ_PUBHEAD, R0                           ; 1241
                                  50             52   D1 00011           CMPL     P, R0                                        ;
                                                 2F   13 00014           BEQL     3$                                           ;
                    24    00F0    C2             02   E0 00016           BBS      #2, 240(P), 2$                               ; 1242
                                  1F      00F0   C2   E8 0001C           BLBS     240(P), 2$                                   ; 1243
                            04    AC             52   D1 00021           CMPL     P, NEW_PUB                                   ; 1244
                                                 19   13 00025           BEQL     2$                                           ;
                                          04     A3   9F 00027           PUSHAB   4(R3)                                        ; 1252
                                                 11   DD 0002A           PUSHL    #17                                          ;
                                                 52   DD 0002C           PUSHL    P                                            ;
                          FEDC    CF             03   FB 0002E           CALLS    #3, PHN$SEND_SMB                             ;
                                          10     A2   9F 00033           PUSHAB   16(P)                                        ; 1253
                                                 11   DD 00036           PUSHL    #17                                          ;
                                          04     AC   DD 00038           PUSHL    NEW_PUB                                      ;
                          FECF    CF             03   FB 0003B           CALLS    #3, PHN$SEND_SMB                             ;
                                  52             62   D0 00040 2$:       MOVL     (P), P                                       ; 1256
                                                 C7   11 00043           BRB      1$                                           ; 1241
                                                      04 00045 3$:       RET                                                   ; 1261

; Routine Size:  70 bytes,    Routine Base:  $CODE$ + 0671
```

```
1 %sbttl 'PHN$FORCED_LINK - Handle Forced Link Message'
1 !++
1 ! Functional Description:
1 !         This steering message routine handles the forced_link message,
1 !         which someone sends us when they want to force us to establish
1 !         a link to a third party.  This is done by the person responsible
1 !         for setting up a conference call.
1 !
1 ! Formal Parameters:
1 !         from_msg        The address of a descriptor of the message.  It
1 !                         consists, as usual, of the sender's node/user name
1 !                         spec followed by an eofrom character.  Following this
1 !                         is the node/user name spec of the person we are being
1 !                         forced to link to.
1 !
1 ! Implicit Inputs:
1 !         global data
1 !
1 ! Implicit Outputs:
1 !         global data
1 !
1 ! Returned Value:
1 !         none
1 !
1 ! Side Effects:
1 !
1 !--
1
1
2 global routine phn$forced_link(from_msg): novalue = begin
2
2 bind
2         from_msg_dsc = .from_msg: descriptor;
2
2 local
2         status: long,
2         third_party_dsc: descriptor,
2         sender_tsb: tsb,
2         third_party_tsb: tsb,
2         tp: ref pub,                            ! Pointer to third party's PUB.
2         p: ref pub;
2
2
2 ! We begin by rebuilding the from_msg descriptor so that it only describes
2 ! the sender's spec.  We build a new descriptor, third_party_dsc, to describe
2 ! the spec of the third party.
2
2 third_party_dsc[ptr] = ch$find_ch(.from_msg_dsc[len],.from_msg_dsc[ptr],
2                                   eofrom) + 1;
3 third_party_dsc[len] = .from_msg_dsc[len] - (.third_party_dsc[ptr] -
2                                              .from_msg_dsc[ptr]);
2 from_msg_dsc[len] = .from_msg_dsc[len] - .third_party_dsc[len] - 1;
2
2 ! Now we make TSBs for both the sender and the third party, because we need
2 ! to parse both of their specs.
2
2 status = phn$make_tsb(from_msg_dsc,sender_tsb);
```

```
; 1003          1319  2 check (.status);
; 1004          1320  2 status = phn$make_tsb(third_party_dsc,third_party_tsb);
; 1005          1321  2 check (.status);
; 1006          1322  2
; 1007          1323  2 ! Now we have to scan the PUB chain and make sure that we do not already
; 1008          1324  2 ! have a link to the third party.  If so, just ignore the message.
; 1009          1325  2
; 1010          1326  2 p = .phn$gq_pubhead[0];
; 1011          1327  3 until .p eqla phn$gq_pubhead do (
; 1012          1328  3         if phn$cmp_target(p[pub_b_tsb],third_party_tsb) then
; 1013          1329  3                 return;
; 1014          1330  3         p = .p[pub_l_flink];
; 1015          1331  2 );
; 1016          1332  2
; 1017          1333  2 ! Now we establish a link to the third party.  We have to remember to
; 1018          1334  2 ! make their PUB permanent, and to assign them a viewport.  If anything
; 1019          1335  2 ! prevents this, just bag the link.
; 1020          1336  2
; 1021          1337  2 status = phn$estab_link(third_party_tsb[tsb_q_tkndsc,0],tp);
; 1022          1338  2 if .status nequ phn$_ok then
; 1023          1339  2         return;
; 1024          1340  2 tp[pub_v_temporary] = false;
; 1025          1341  2 status = phn$fresh_screen(false);
; 1026          1342  3 if .status nequ phn$_ok then (
; 1027          1343  3         phn$break_link(.tp,smb__hungup);
; 1028          1344  3         return;
; 1029          1345  2 );
; 1030          1346  2
; 1031          1347  2 ! Finally, inform the user about the third party, including the sender's
; 1032          1348  2 ! name and the third party's name.
; 1033          1349  2
; 1034          1350  2 phn$inform(phn$_confcall,
; 1035          1351  2                 sender_tsb[tsb_q_tkndsc,.sender_tsb      [tsb_w_tkncount]],
; 1036          1352  2           third_party_tsb[tsb_q_tkndsc,.third_party_tsb[tsb_w_tkncount]]);
; 1037          1353  2 return;
; 1038          1354  2
; 1039          1355  1 end;




                                        003C 00000          .ENTRY  PHN$FORCED_LINK, Save R2,R3,R4,R5            ; 1291
                   55 00000000G    8F   D0 00002          MOVL    #PHN$_OK, R5                                  ;
                   54 00000000G    00   9E 00009          MOVAB   LIB$SIGNAL, R4                                ;
                   5E      FE2C    CE   9E 00010          MOVAB   -468(SP), SP                                  ;
                   52        04    AC   D0 00015          MOVL    FROM_MSG, R2                                  ; 1294
     04   B2       62              00   3A 00019          LOCC    #0, (R2), @4(R2)                              ; 1309
                                   02   12 0001E          BNEQ    1$                                            ;
                                   51   D4 00020          CLRL    R1                                            ;
             FC    AD        01    A1   9E 00022 1$:      MOVAB   1(R1), THIRD_PARTY_DSC+4                      ; 1310
     50      04    A2        FC    AD   C3 00027          SUBL3   THIRD_PARTY_DSC+4, 4(R2), R0                  ; 1312
 F8  AD            50              62   A1 0002D          ADDW3   (R2), R0, THIRD_PARTY_DSC                     ; 1311
                   50              62   3C 00032          MOVZWL  (R2), R0                                      ; 1313
                   51        F8    AD   3C 00035          MOVZWL  THIRD_PARTY_DSC, R1                           ;
                   50              51   C2 00039          SUBL2   R1, R0                                        ;
          62       50              01   A3 0003C          SUBW3   #1, R0, (R2)                                  ;
```

```
                          00E8    CE   9F 00040          PUSHAB  SENDER_TSB                                  ; 1318
                                  52   DD 00044          PUSHL   R2                                          ;
               0000G  CF          02   FB 00046          CALLS   #2, PHN$MAKE_TSB                            ;
                      53          50   D0 0004B          MOVL    R0, STATUS                                  ;
                      05          53   E8 0004E          BLBS    STATUS, 2$                                  ; 1319
                                  53   DD 00051          PUSHL   STATUS                                      ;
                      64          01   FB 00053          CALLS   #1, LIB$SIGNAL                              ;
                            04    AE   9F 00056 2$:      PUSHAB  THIRD_PARTY_TSB                             ; 1320
                            F8    AD   9F 00059          PUSHAB  THIRD_PARTY_DSC                             ;
               0000G  CF          02   FB 0005C          CALLS   #2, PHN$MAKE_TSB                            ;
                      53          50   D0 00061          MOVL    R0, STATUS                                  ;
                      05          53   E8 00064          BLBS    STATUS, 3$                                  ; 1321
                                  53   DD 00067          PUSHL   STATUS                                      ;
                      64          01   FB 00069          CALLS   #1, LIB$SIGNAL                              ;
                      52   0000G  CF   D0 0006C 3$:      MOVL    PHN$GQ_PUBHEAD, P                           ; 1326
                      50   0000G  CF   9E 00071 4$:      MOVAB   PHN$GQ_PUBHEAD, R0                          ; 1327
                      50          52   D1 00076          CMPL    P, R0                                       ;
                                  13   13 00079          BEQL    5$                                          ;
                            04    AE   9F 0007B          PUSHAB  THIRD_PARTY_TSB                             ; 1328
                            0C    A2   9F 0007E          PUSHAB  12(P)                                       ;
               0000G  CF          02   FB 00081          CALLS   #2, PHN$CMP_TARGET                          ;
                      55          50   E8 00086          BLBS    R0, 7$                                      ;
                      52          62   D0 00089          MOVL    (P), P                                      ; 1330
                                  E3   11 0008C          BRB     4$                                          ; 1327
                                  5E   DD 0008E 5$:      PUSHL   SP                                          ; 1337
                            0C    AE   9F 00090          PUSHAB  THIRD_PARTY_TSB+4                           ;
               F948   CF          02   FB 00093          CALLS   #2, PHN$ESTAB_LINK                          ;
                      53          50   D0 00098          MOVL    R0, STATUS                                  ;
                      55          53   D1 0009B          CMPL    STATUS, R5                                  ; 1338
                                  3E   12 0009E          BNEQ    7$                                          ;
                      52          6E   D0 000A0          MOVL    TP, R2                                      ; 1340
               00F0   C2          04   8A 000A3          BICB2   #4, 240(R2)                                 ;
                                  7E   D4 000A8          CLRL    -(SP)                                       ; 1341
               0000G  CF          01   FB 000AA          CALLS   #1, PHN$FRESH_SCREEN                        ;
                      53          50   D0 000AF          MOVL    R0, STATUS                                  ;
                      55          53   D1 000B2          CMPL    STATUS, R5                                  ; 1342
                                  0A   13 000B5          BEQL    6$                                          ;
                                  09   DD 000B7          PUSHL   #9                                          ; 1343
                                  52   DD 000B9          PUSHL   R2                                          ;
               FDA7   CF          02   FB 000BB          CALLS   #2, PHN$BREAK_LINK                          ;
                                       04 000C0          RET                                                 ; 1342
                      50    06    AE   3C 000C1 6$:      MOVZWL  THIRD_PARTY_TSB+2, R0                       ; 1352
                            08  AE40   7F 000C5          PUSHAQ  THIRD_PARTY_TSB+4[R0]                       ;
                      50  FF16    CD   3C 000C9          MOVZWL  SENDER_TSB+2, R0                            ; 1351
                          FF18  CD40   7F 000CE          PUSHAQ  SENDER_TSB+4[R0]                            ;
                        00000000G  8F  DD 000D3          PUSHL   #PHN$_CONFCALL                              ; 1352
               0000G  CF          03   FB 000D9          CALLS   #3, PHN$INFORM                              ;
                                       04 000DE 7$:      RET                                                 ; 1355

; Routine Size:  223 bytes,    Routine Base:  $CODE$ + 06B7


; 1040          1356  1
; 1041          1357  0 end eludom
```

```
                                                                                    .EXTRN  LIB$SIGNAL

;                                         PSECT SUMMARY
;
;      Name                        Bytes                          Attributes
;
;  $PLIT$                             272  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                              316  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                            1942  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                              Library Statistics
;
;                                              -------- Symbols --------       Pages        Processing
;      File                                    Total    Loaded   Percent       Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1            9776        47         0         581          00:00.7



;                                          COMMAND QUALIFIERS

;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LINKSUBS/OBJ=OBJ$:LINKSUBS MSRC$:LINKSUBS/UPDATE=(ENH$:LINKSUBS)

; Size:           1942 code + 588 data bytes
; Run Time:          00:26.7
; Elapsed Time:      01:36.1
; Lines/CPU Min:      3044
; Lexemes/CPU-Min:  32221
; Memory Used:  298 pages
; Compilation Complete
```

0305 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY